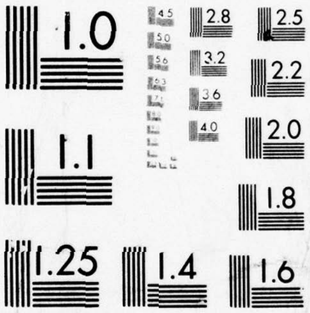
AD-A040 081 NAVAL RESEARCH LAB WASHINGTON D C F/G 17/9
A METHOD FOR COMPUTING VERTICAL-PLANE COVERAGE DIAGRAMS FOR FRE--ETC(U)
APR 77 W H HARPER, P N MARINOS

UNCLASSIFIED NRL-MR-3475 NL

| OF |
AD
A040 081

END
DATE
FILMED
6-77


1.0
4.5
5.0
5.6
6.3
7.1
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

12

NRL Memorandum Report 3475

A Method for Computing Vertical-Plane Coverage Diagrams for Frequency Agile Pulse Radar Systems

W. H. Harper and Peter N. Marinos

Search Radar Branch
Radar Division

April 1977

NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release; distribution unlimited.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(14) REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL-MR-3475 ~~NRL Memorandum Report 3475~~	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) A METHOD FOR COMPUTING VERTICAL-PLANE COVERAGE DIAGRAMS FOR FREQUENCY AGILE PULSE RADAR SYSTEMS.		5. TYPE OF REPORT & PERIOD COVERED (9) Final report ~~on an NRL problem.~~
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) W.H. Harper [redacted] P.N. Marinos		8. CONTRACT OR GRANT NUMBER(s) 62712N
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, D.C. 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NRL Problem R02-92 Project ZF-12-151-001
11. CONTROLLING OFFICE NAME AND ADDRESS Department of the Navy Chief of Naval Material Washington, D.C. 20360		12. REPORT DATE (11) April 1977
		13. NUMBER OF PAGES ~~52~~ (12) 51p.
14. MONITORING ~~AGENCY NAME & ADDRESS~~ (if different from Controlling Office) (16) F12151 (17) ZF12151001		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION ~~STATEMENT~~ (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer program
Radar vertical-plane coverage diagrams
Frequency agile radars
Computer plotting

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A method for obtaining computer plots of vertical-plane coverage diagrams (i.e., range-height-angle charts) for frequency agile pulse radar systems is presented. The approach is based on well-known monochromatic radar detection calculations which are here extended to cover the frequency agile case.

251 95Ø

CONTENTS

A METHOD FOR COMPUTING VERTICAL-PLANE COVERAGE DIAGRAMS FOR FREQUENCY AGILE PULSE RADAR SYSTEMS

INTRODUCTION

In a comprehensive report on maximum radar range computations, Blake [1] provides all the necessary tools for obtaining coverage plots (i.e., range-height-angle charts) of monochromatic pulse radar systems. This paper provides a method for extending the range computations applicable to the monochromatic case to pulse radar systems featuring frequency agility. The number of different frequencies employed, as well as the number of consecutive pulses to be transmitted at the same frequency, are both arbitrary.

MATHEMATICAL ANALYSIS

The detection scheme assumed in the analysis which follows utilizes a square-law detector followed by a linear integrator, as shown in Fig. 1. The symbols D_o, X_o, and Y_o denote the SNR (i.e., signal-to-noise power ratio) at the designated points of the detector-integrator complex.

Based on approximate closed-form expressions derived by Barton [2] and later simplified by Cann [3], one may relate the input and output SNRs using the equation

$$X_o = \frac{2D_o^2}{D_o + 2.3} \qquad (1)$$

The integrated output SNR, Y_o, may now be written in the form

$$Y_o = \sum_{j=1}^{N} X_{oj} = \sum_{j=1}^{N} \left\{ \frac{2D_{oj}^2}{D_{oj} + 2.3} \right\} \qquad (2)$$

where N denotes the number of integrated pulses and D_{oj} represents the SNR at the detector input due to the jth pulse illuminating a target located at R_{max}; it is assumed that R_{max} represents the maximum detection range associated with the total energy delivered on the target by the N transmitted pulses.

Since our objective is to relate the integrated SNR (i.e., Y_o) to a maximum detection range, R_{max}, using the classical radar equation, an average SNR is defined at the detector output in the form

Note: Manuscript submitted March 4, 1977.

$$\hat{X}_o = \frac{1}{N} \sum_{j=1}^{N} X_{oj} = \frac{1}{N} \sum_{j=1}^{N} \frac{2D_{oj}^{2}}{D_{oj} + 2.3} \qquad (3)$$

The SNR given by Eq. (3) may be thought of as the SNR necessary to achieve a specified probability of detection, P_d, and false alarm, P_{fa}, for a specified number of pulses and a postulated type of target; normally, it is the corresponding SNR, $\hat{D}_o$, at the input of the square-law detector rather than $\hat{X}_o$ that one associates with P_d and P_{fa}. Given, however, $\hat{D}_o$, the corresponding value of $\hat{X}_o$ is obtained using Eq. (1). For example, given N, P_d, and P_{fa}, and a target described by Swerling's Case 2, one may utilize published results [1] to determine $\hat{D}_o$ and subsequently X_o.

Once $\hat{X}_o$ is obtained, the problem becomes one of relating $\hat{X}_o$ to a corresponding maximum detection range, R_{max}, in a way which takes advantage of the computational aids developed by Blake [1,4]. This is accomplished by writing the radar equation in the form,

$$D_{oj} = \frac{P_t A^2 \sigma F_j^{\,4}}{4\pi\lambda_j^{\,2}\, P_{noise} \cdot R_{max}^{\,4}} = \left[\frac{P_t A^2 \sigma}{4\pi\lambda_j^{\,2}\, \hat{D}_o\, P_{noise}} \right] F_j^{\,4} \cdot \frac{\hat{D}_o}{R_{max}^{\,4}} \qquad (4)$$

where D_{oj} represents the SNR at the detector input due to the jth pulse return from a target positioned at R_{max}; $\hat{D}_o$ is the (averaged) minimum SNR based on N pulses; F_j denotes the pattern-propagation factor; and the remaining symbols in Eq. (4) represent well-known radar parameters [5]. The bracketed portion of Eq. (4) represents the free-space range, $R_{jfs}^{\,4}$, corresponding to N pulses, but as if they all had been transmitted at frequency f_j (i.e., wavelength λ_j) and each resulted in a SNR, $\hat{D}_o$, at the detector input. In view of the above interpretation, one may rewrite Eq. (4) in the form,

$$D_{oj} = \hat{D}_o \left(\frac{R_{xj}}{R_{max}} \right)^4 \qquad (5)$$

where $R_{xj} = R_{jfs}\, F_j$, and $j = 1,2,\ldots,N$.

Since

$$R_{jfs}^{\,4} = \frac{P_t A^2 \sigma}{4\pi\lambda_j^{\,2}\, \hat{D}_o\, P_{noise}}\,, \qquad (6)$$

varying only the frequency results in a relationship of the form

$$R_{jfs}^{\,4} = R_{ifs}^{\,4} \left(\frac{\lambda_i}{\lambda_j} \right)^2 = R_{ifs}^{\,4} \left(\frac{f_j}{f_i} \right)^2 \qquad (7)$$

which provides the free-space range at frequency f_j in terms of the free-space range at another frequency, f_i, while all other parameters in Eq. (6) remain fixed.

In some instances the maximum free-space range, $R_{j\,max}$, based on the number of pulses at each frequency, is known. In this case, there is a $\hat{D}_{oj}$ given by

$$\hat{D}_{oj} = \frac{P_t A^2 \sigma F_j^{\,4}}{4\pi\lambda_j^{\,2}\, P_{noise}\, R_{j\,max}^{\,4}} \quad . \tag{8}$$

The free-space range at frequency f_j is then given by

$$R_{jfs}^{\,4} = R_{j\,max}^{\,4}\left(\frac{\hat{D}_{oj}}{\hat{D}_o}\right) \tag{9}$$

Substituting next Eqs. (3) and (5) into Eq. (1), that is, into expression

$$\hat{X}_o = \frac{2\hat{D}_o}{1 + 2.3\left(\frac{1}{\hat{D}_o}\right)} \tag{10}$$

one obtains

$$\frac{1}{1+\alpha} = \frac{1}{N}\sum_{j=1}^{N}\left[\frac{\left(\frac{R_{xj}}{R_{max}}\right)^4}{1+\alpha\left(\frac{R_{max}}{R_{xj}}\right)^4}\right] \tag{11}$$

where $\alpha = (2.3/\hat{D}_o)$.

PLOTTING TECHNIQUES

Solution of Eq. (11) for R_{max} provides the information necessary to produce the so-called radar vertical-plane coverage diagrams.

Blake [4] has developed a computer program for presenting the radar interference lobing phenomena on range-height-angle plots. This program was named LOBEPLOT and was written in Fortran for use on a CDC-3800 computer.

Relatively simple modifications to LOBEPLOT have been made which have resulted in the solution of Eq. (11) and enabled the computer (Calcomp) plotting of the lobing charts for frequency agile radars. The original modifications were made, and the program was debugged on a CDC-3800 computer at NRL. More recently, the program has been changed to allow running on the new Texas Instruments Advanced Scientific Computer (ASC) at NRL. This new version has been named LOBMUF for "lobes, multi-frequency." The complete program is listed in Appendix A; the input data card formats are given in Appendix B.

In LOBMUF, P_d, P_{fa}, the total number of pulses integrated, the Swerling fluctuation case, and the number of radars or radar frequencies are specified on input data cards. In addition, for each radar or radar frequency, the free-space range, beamwidth, number of pulses at that frequency, sidelobe level, and a parameter FREF are required. FREF is zero if the free-space range was calculated at each frequency as indicated in connection with Eqs. (8) and (9); otherwise, FREF is the midband or average frequency and the free-space range corresponding to this reference frequency. Other inputs to the program are identical to those for LOBEPLOT. They concern the dimensions of the range-height chart, polarization radiated, antenna height and tilt, etc.

Using subroutines from another of Blake's computer programs [5], RGCALC, the signal-to-noise ratio, $\hat{D}_o$, and the parameter, α, are computed. If FREF = 0, $\hat{D}_{oj}$ for each frequency is calculated. Next, the pattern propagation factor, F_j, is calculated exactly as in Blake's LOBEPLOT. Finally, a simple search routine is used to converge on the value of R_{max} in Eq. (11). The search is terminated when the two sides of Eq. (11) differ by less than 0.01.

Figure 2 gives a typical coverage diagram generated by the computer program LOBMUF. The inputs involved a hypothetical radar radiating two pulses at each of 21 frequencies from 1350 MHz to 1650 MHz in 15 MHz steps. 1500 MHz was chosen as the reference frequency, and the free-space range was chosen to be 100 n. mi.

Figure 3 shows the lobing plot for a single frequency radar at 1500 MHz. A comparison of Figs. 2 and 3 shows that the incoherent integration in a frequency agile radar does much to fill in the nulls of the lobing pattern and give a solid elevation coverage.

SUMMARY

A computer program has been developed to plot range-height-angle lobing charts for frequency agile radars. This program is quite flexible. It may be used to plot the lobing charts for several radars or a single radar including several frequencies. This program, LOBMUF, builds on Blake's range-height charts and lobing plots. The program assumes incoherent integration and a closed form expression for a square-law detector.

ACKNOWLEDGMENT

Thanks are due to Dr. W. M. Waters of NRL for many helfpul discussions concerning this problem.

REFERENCES

1. L. V. Blake, "A Guide to Basic Pulse Radar Maximum-Range Calculation; Part 1 - Equations, Definitions, and Aids to Calculation," NRL Report 6930, Dec 23, 1969

2. D. K. Barton, "Simple Procedures for Radar Detection Calculations," IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-5, pp 837-846, Sep 1969

3. A. J. Cann, "Simple Radar Detection Calculation," IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-8, pp 73-74, Jan 1972

4. L. V. Blake, "Machine Plotting of Radio/Radar Vertical-Plane Coverage Diagrams," NRL Report 7098, Jun 28, 1970

5. L. V. Blake, "A Fortran Computer Program to Calculate the Range of a Pulse Radar," NRL Report 7448, Aug 28, 1972.

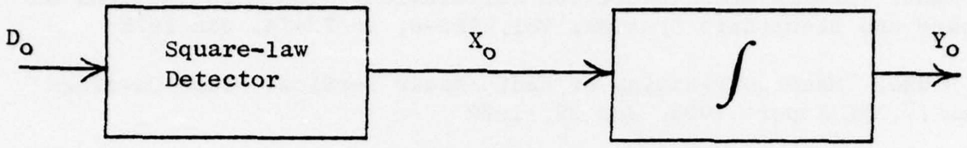


Fig. 1 — Radar detection and integration diagram

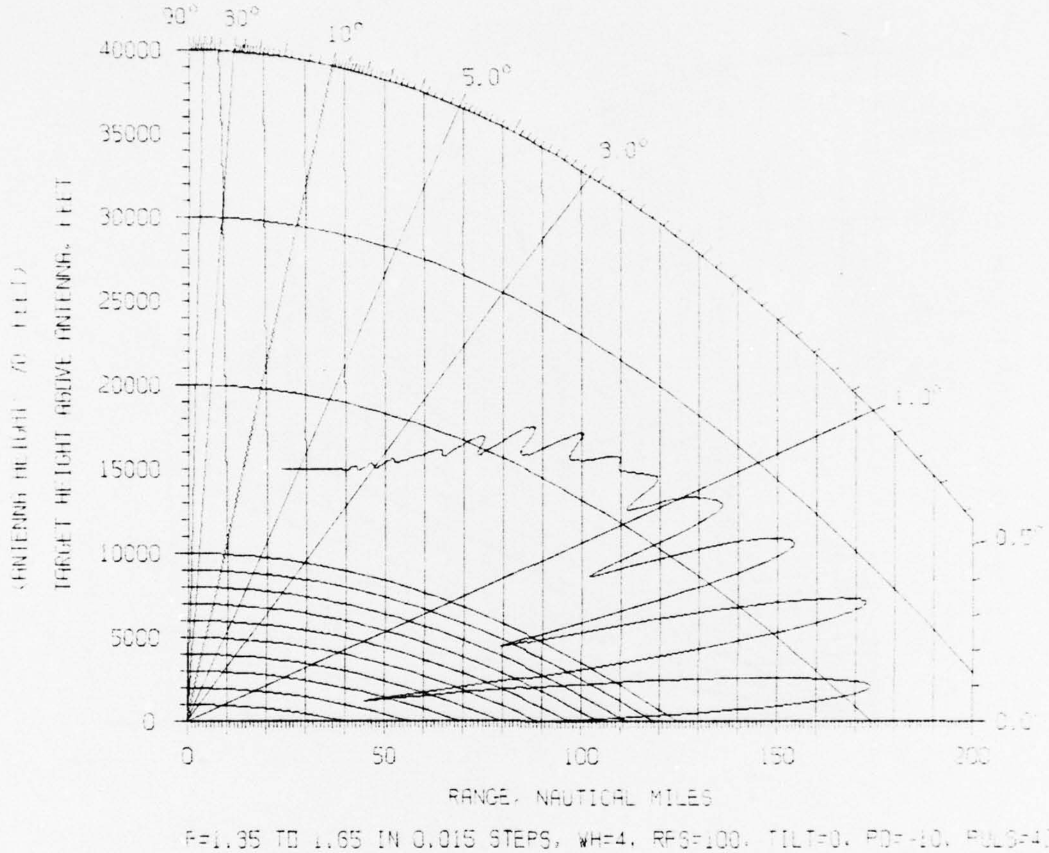


Fig. 2 — Lobing chart for hypothetical radar radiating two pulses at each of 21 frequencies from 1350 MHz to 1650 MHz in 15 MHz steps. Wave height of 4 feet, free-space range of 100 n. mı., and vertical beamwidth of 4.0° were assumed.

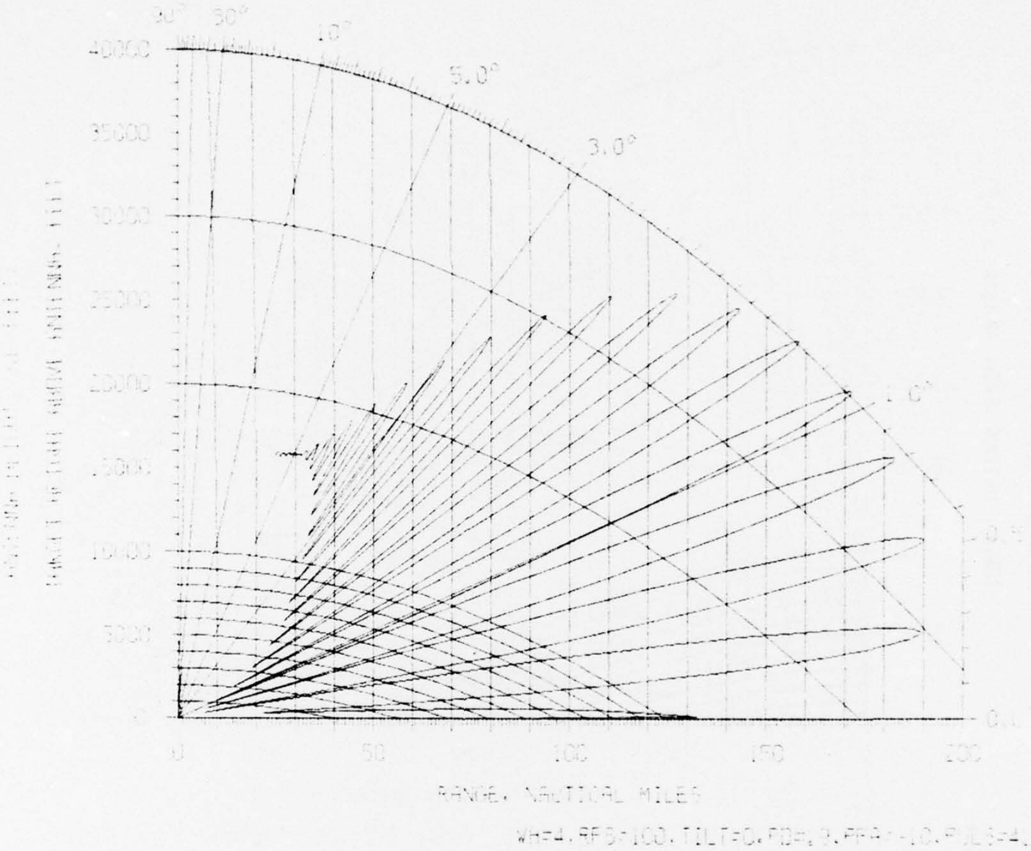


Fig. 3 — Lobing chart for single frequency radar at 1500 MHz with free-space range of 100 n. mi. A wave height of 4 feet and vertical beamwidth of 4.0° were assumed.

APPENDIX A: Fortran Listing for Program LOBMUF

The basic modifications to Blake's LOBEPLOT for program LOBMUF occur in Subroutine LOBES, as described in the Plotting Techniques section of this report. In addition, Subroutines PDSN, PD, INVERS, MARSWR, DGAM, DEVAL, GAM, EVAL, and SUMLOG are incorporated directly from RGCALC.

```
      STATEMENT


      PRØGRAM LØBMUF
      REAL*8 ILABL(10)
      CØMMØN /MTPE/ Q1,Q2,Q3,Q4,Q5,Q6,ERRØR
      CØMMØN/ADPLTS/IADD
      CØMMØN/CSC/ICSC
      CØMMØN/LT/LTN
      DIMENSIØN A(1500)
      DATA LTN/0/
      CALL RSSTØP
      READ 1,ANTHGT
      ERRØR=0.001
      CALL PLØTS (A,1500,0.5)
      PRINT 30
   30 FØRMAT (' TWØ PEN TURRET PØSITIØNS ARE USED, D10 AND D11.  PLØT ST
     *ARTS WITH PEN TURRET PØSITIØN D10.  CØARSER PEN IS D11.'//)
    2 READ (5,5,END=100) ILABL
  200 READ 3,XMAX,YMAX,RMAX,HMAX,THMIN,THMAX,WHFT,RDR
      NRDR=RDR
      IF ( NRDR .LT. 1 ) GØTØ 110
      SFAC = YMAX*.125
   52 H = .175*SFAC
      Y = 1.5*SFAC
      IF (LTN .EQ. 18) GØ TØ 61
      IF ((YMAX+Y).LE.9.5) GØ TØ 61
   60 SFAC = 9.5/(YMAX+Y)
      YMAX = YMAX*SFAC
      XMAX = XMAX*SFAC
      Y = Y*SFAC
      H = H*SFAC
   61 X=Y+3.*H
      CALL ØRIGIN (X,Y)
      PRINT 10, ILABL
      CALL RHACHT (XMAX,YMAX,RMAX,HMAX,ANTHGT)
      CALL LØBES (XMAX,YMAX,RMAX,HMAX,WHFT,THMIN,THMAX,NRDR)
      CALL LETTER(0.,-Y,H,ILABL,0.0,80)
      CALL ØRIGIN (XMAX+3.*Y,-Y)
      CALL REZERØ
      GØ TØ 2
  100 CALL ENDPLT
      GØTØ 130
  110 PRINT 120,NRDR
  120 FØRMAT(//,' PRØGRAM TERMINATIØN DUE TØ ILLEGAL DATA ENTERED FØR NU
     *MBER ØF RADARS . . . . . NUMBER ENTERED IS  ',I5)
  130 CØNTINUE
    1 FØRMAT (F10.0)
    3 FØRMAT( 8F10.0)
    5 FØRMAT (10A8)
   10 FØRMAT(3X,10A8//)
      END
      STATEMENT

      SUBRØUTINE LØBES (XMAX,YMAX,RMAX,HMAX,WHFT,THMIN,THMAX,NRDR)
C      THIS VERSIØN CØMPLETED JUNE 1976 TØ ALLØW PLØTTING LØBING PATTERN
C     FØR MULTIFREGUENCY CASE WITH INCØHERENT INTEGRATIØN.
      CØMMØN/CSC/ICSC
      CØMMØN /MTPE/  X2, Y2, X11, Y11, XA, YA, ERRØR
      CØMMØN/B/XX(181),YY(181),CT1(181),SN1(181),DELL
      CØMMØN/DUM1/PFF( 2000 )
      DIMENSIØN RXN4(50),NUM(50),SNR(50)
      DIMENSIØN RFS(50),FMHZ(50),BWD(50),SLDB(50)
      DATA IIMAX/2000/
      DATA ICSC / 0/
      DATA PI/3.141592654/
      DATA PI2/6.283185307/
      DATA RDN/.01745329252/
      DATA CØNV/1.645788333E-4/
      DATA AE / 2.786526684E7/
      NRDS=NRDR
      RNØ=FLØAT(NRDS)
      READ 904,PDT,PFA,PULS,CASE,AHFT,TILT,PØL,CSC
      PRINT 905,PDT,PFA,PULS,CASE,AHFT,TILT,PØL,CSC
  905 FØRMAT (/,2X,4HPDT=,F5.2,2X,4HPFA=,F5.2,2X,5HPULS=,F6.1,2X,5HCASE=
     *,F5.2,2X,5HAHFT=,F6.1,2X,5HTILT=,F5.2,2X,4HPØL=,F4.1,2X,4HCSC=,F4.
     *1,/)
      NPULS=PULS
      KASE=CASE
      CALL PDSN (PDT,PFA,NPULS,KASE,SDB)
      SNN=10.**(SDB/10.)
      GAMMA=2.3/SNN
      BETA=1./(1.+GAMMA)
  904 FØRMAT (8F10.0)
      PRINT 590
  590 FØRMAT(' THE CALLING PARAMETERS SENT TØ SUBRØUTINE LØBES ARE AS FØ
     *LLØWS')
      PRINT 591 , XMAX , YMAX , RMAX , HMAX , WHFT , THMIN , THMAX, NRDR
  591 FØRMAT(2X,5HXMAX=,F4.1,2X,5HYMAX=,F4.1,2X,5HRMAX=,F6.1,2X,5HHMAX=,
     * F9.1,2X,5HWHFT=,F5.1,2X,6HTHMIN=,F4.1,2X,6HTHMAX=,F4.1,2X,5HNRDR=
     *,I3)
      DØ 84 II=1,IIMAX
84    PFF(II) = 0.0
      E = YMAX*RMAX/(XMAX*HMAX*CØNV)
      EX=XMAX/RMAX
      Z=WHFT*.3535534
      XP=1.E-45
      YP=1.E-45
      DØ 801 JJ=1,NRDS
      READ (5,802,END=807) RANGE,FREQ,BEAM,SDLØB,PULNUM,FREF
  803 ICSC=CSC
      IPØL=PØL
      NUM(JJ)=PULNUM
      RFS(JJ)=RANGE
      FMHZ(JJ)=FREQ
      STATEMENT


      BWD(JJ)=BEAM
      SLDB(JJ)=SDLDB
  802 FORMAT (6F10.0)
      PRINT 499 , NRDR
  499 FORMAT(//,20X,'THE RADAR PARAMETERS INPUT FOR RADAR NUMBER  ',I3
     *,'  ARE AS FOLLOWS')
      PRINT 592,RFS(JJ),AHFT,FMHZ(JJ),BWD(JJ),SLDB(JJ),TILT,POL,CSC,PULN
     *UM,FREF
  592 FORMAT(/,2X,4HRFS=,F6.1,2X,5HAHFT=,F6.1,2X,5HFMHZ=,F7.1,2X,4HBWD=,
     *F5.1,2X,5HSLDB=,F5.1,2X,5HTILT=,F5.1,2X,4HPOL=,F2.0,2X,4HCSC=,F2.0
     *,2X,6HNPULS=,F5.0,2X,5HFREF=,F7.1/)
      IF (FREF.NE.0.) GO TO 514
      CALL PDSN (PDT,PFA,NUM(JJ),KASE,SDB)
      SNR(JJ)=10.**(SDB/10.)
      GO TO 515
  514 SNR(JJ)=SNN
      RFS(JJ)=RFS(JJ)*SQRT(SQRT(FMHZ(JJ)/FREF))
  515 CONTINUE
      RFS(JJ)=RFS(JJ)*(SQRT(SQRT(SNR(JJ)/SNN)))
      NRDR=NRDR-1
  801 CONTINUE
      TILTR = TILT * RDN
      THMINR=THMIN*RDN
      THMAXR=THMAX*RDN
      DEL1= ( THMAXR - THMINR ) / IIMAX
       PI23DE = 3.0*DEL1 + PI2
      THET2=THMINR-DEL1
      N = 0
      INDEX = 0
      F=1.5
      IDASH=1
      CALL DASHON
      AH2=AHFT*AHFT
      HAE = 2. * AHFT/(3.*AE)
      AEH = AE * (AE + AHFT)
      PARAM = SQRT (AE/(2.*AHFT))
      DO 810  II= 1,IIMAX
      PF4=0.
      THET2 = THET2 + DEL1
      IF (THET2 .GE. PI23DE      ) GOTO 40
      T2 = TAN (THET2)
      S2 = SIN(THET2)
      S3 = S2
      PSI = THET2
      DO 805 M=1,NRDS
      BWR=BWD(M)*RDN
      BW2 = BWR*.5
      CSCT = .7071 * SIN(BW2+TILTR)
      IF (BWD(M).LE.45.) GO TO 251
  250 IBEAM=0
      CONST=90./BWD(M)
```

```
    STATEMENT


    IF (BWD(M).EQ.360.) IBEAM=-1
    GO TO 252
251 IBEAM=1
    CONST=1.39157/SIN(BW2)
    ANLL =ARSIN(PI/CONST)
    CPAT1=10.**((13.2-SLDB(M))*.05)
252 W = 983.573/FMHZ(M)
    W4 = 0.25 * W
    WLIM = 0.01 * W
    FAC = PI2/W
    PD = 2. * AHFT * S2
    IF (INDEX .EQ. 1) GO TO 77
    T23 = T2/ 3.
    GAM = HAE/(SQRT (T23**2+HAE)+T23)
    PSI = THET2 + GAM
    S3 = SIN(PSI)
    ZETA = PARAM * T2
    D1=0.57735*SQRT(1.+2.*ZETA/SQRT(ZETA*ZETA+3))
    PD1=SQRT(AH2+AEH*GAM*GAM)*2.*S3*S3
    IF (ABS(PD-PD1).GE.WLIM) GO TO 79
 78 IF (D1.LT.0.999) GO TO 79
780 D1 = 1.
    INDEX = 1
 79 PD = PD1
    IF (PD.LT.W4.OR.THET2.LT.0.000873) GO TO 300
    GO TO 301
301 IF (IDASH.NE.1) GO TO 300
302 CALL DASHOF
    IDASH=0
300 CONTINUE
 77 CALL SEAREF (FMHZ(M),PSI,IPOL,RHO,PHI)
    PIZ=PI*Z*S3/W
    PIZ2=PIZ*PIZ
    RUF=EXP(-8.*PIZ2)
    ANG = THET2 - TILTR
    IF(IBEAM) 260,261,262
260 PAT=1.
    GO TO 24
261 PAT=COS(CONST*ANG)
    GO TO 24
262 IF (ICSC.NE.1) GO TO 61
 80 IF (ANG.LE.BW2) GO TO 61
 60 PAT = CSCT/S2
    GO TO 24
 61 IF (ANG.NE.0.) GO TO 23
 22 PAT = 1.
    GO TO 24
 23 UU = CONST * SIN (ANG)
    INT = UU/PI2
    DIFF = UU - INT * PI2
    IF (ABS(ANG).LE.ANLL) GO TO 161
    STATEMENT

160 CPAT = CPAT1
    GO TO 162
161 CPAT = 1.
162 PAT = CPAT*SIN(DIFF)/UU
 24 ANGR = THET2 + TILTR + 2. * GAM
    IF (IBEAM) 270,271,272
270 PATR=1.
    GO TO 27
271 PATR=COS(CONST*ANGR)
    GO TO 27
272 IF (ANGR.NE.0.) GO TO 26
 25 PATR = 1.
    GO TO 27
 26 UUR = CONST * SIN(ANGR)
    INTR = UUR / PI2
    DIFFR = UUR - INTR * PI2
    IF (ABS(ANGR).LE.ANLL) GO TO 66
 65 CPAT = CPAT1
    GO TO 67
 66 CPAT = 1.
 67 PATR = CPAT*SIN(DIFFR)/UUR
 27 IF (ABS(PAT).GE.1.E-44) GO TO 29
 28 IF (PAT.GE.0.) GO TO 281
280 D = -RHO*RUF*PATR*1.E45*D1
    GO TO 30
281 D=D1*RHO*RUF*PATR*1.E45
    GO TO 30
 29 D = D1* RHO * PATR/PAT * RUF
 30 ALPHA = FAC * PD + PHI
    INT1 = ALPHA/PI2
    DIFF1 = ALPHA- INT1*PI2
    F = ABS(PAT)* SQRT(1.0 + D*D + 2.0*D*COS(DIFF1))
    RXN4(M)=(RFS(M)*F)**4
    PF4=PF4+SQRT(SQRT(RXN4(M)))/RNO
805 CONTINUE
    RNG=PF4
    RNG2=RNG*RNG
    RNG4=RNG2*RNG2
    FACMIN=10.
997 SUMEN=0.
    DO 998 MM=1,NRDS
    RRATIO=RXN4(MM)/RNG4
    PNO=FLOAT(NUM(MM))
    SUMEN=(PNO/PULS)*(RRATIO/(1.+GAMMA/RRATIO))+SUMEN
998 CONTINUE
    FACMN2=FACMIN
    FACMIN=SUMEN-BETA
    FSUM=FACMN2+FACMIN
    IF (RNG.EQ.0.) GO TO 993
    IF (ABS (FACMIN).LT.0.01) GO TO 993
    IF (FACMIN.GE.-1.) GO TO 902
    STATEMENT


901 HØNUM=AINT(FACMIN)
    FACMIN=FACMIN-HØNUM
902 CØNTINUE
    RPREV=RNG
    RNG4=RNG4*(1.+FACMIN)
    RNG=SQRT(SQRT(RNG4))
    RDIFF=ABS (RPREV-RNG)
    IF (RDIFF.LT.0.1.AND.FSUM.LT.0.01) GØ TØ 993
    GØ TØ 997
993 CØNTINUE
999 PFF(II)=RNG
810 CØNTINUE
    THET2=THMINR-DEL1
    DØ 806 II=1,IIMAX
    A=PFF(II)*EX
    THET2 = THET2 + DEL1
    N=N+1
    X = A * CØS(THET2)
    Y = A * E * SIN(THET2)
 45 IF (X.LE.XMAX) GØ TØ 501
500 ISUMX=8
    GØ TØ 502
501 ISUMX=0
502 IF (Y.LE.YMAX) GØ TØ 504
503 ISUMY=4
    GØ TØ 505
504 ISUMY=0
505 IF (XP.LE.XMAX) GØ TØ 507
506 ISUMXP=2
    GØ TØ 508
507 ISUMXP=0
508 IF (YP.LE.YMAX) GØ TØ 510
509 ISUMYP=1
    GØ TØ 511
510 ISUMYP=0
511 ISUM=ISUMX+ISUMY+ISUMXP+ISUMYP+1
    GØ TØ (601,602,603,602,605,620,620,620,609,620,620,620,605,620),
   *ISUM
601 IF (N.NE.1) GØ TØ 3
  2 CALL PLØT (X,Y,3)
    X2=X
    Y2=Y
    XP=X
    YP=Y
    GØTØ 806
  3 IF (N.NE.2) GØ TØ 440
 44 XA=X
    YA=Y
    XP=X
    YP=Y
    X11=X
    STATEMENT


    Y11=Y
    GOTO 806
440 CALL MINTAP (X,Y)
620 XP=X
    YP=Y
    IF(THET2.GE.THMAXR) GO TO 40
    GOTO 806
602 IF (N.EQ.1) GO TO 620
    CALL DASHON
    IDASH=0
    IF (IAXIS.EQ.1) CALL PLOT(XMAX,YMAX,2)
    CALL INTRST  (0.,  YMAX,XMAX,YMAX,X,Y,XP,YP,X0,Y0)
621 CALL PLOT(X0,Y0,2)
    CALL DASHOF
    X2=X0
    Y2=Y0
    XA=X
    X11=X
    XP=X
    YA=Y
    Y11=Y
    YP=Y
    N=2
    GOTO 806
603 IF (N.EQ.1) GO TO 620
    CALL DASHON
    IDASH=0
    CALL INTRST (XMAX,0.,XMAX,YMAX,X,Y,XP,YP,X0,Y0)
    GO TO 621
605 IF (N.EQ.1) GO TO 620
    CALL MINTAP (XP,YP)
    CALL PLOT(XP,YP,2)
    CALL INTRST (0.,YMAX,XMAX,YMAX,X,Y,XP,YP,X0,Y0)
    IAXIS=2
    CALL PLOT(X0,Y0,2)
    GO TO 620
609 IF (N.EQ.1) GO TO 620
    CALL MINTAP (XP,YP)
    CALL PLOT(XP,YP,2)
    CALL INTRST (XMAX,0.,XMAX,YMAX,X,Y,XP,YP,X0,Y0)
    IAXIS=1
    CALL PLOT(X0,Y0,2)
    GO TO 620
 40 IF (ISUM.NE.1) GO TO 651
650 CALL PLOT(X,Y,2)
    RETURN
651 IF (X.GT.XMAX.AND.XP.GT.XMAX) GO TO 652
    GO TO 653
652 CALL INTRST (XMAX,0.,XMAX,YMAX,0.,0.,X,Y,X0,Y0)
    CALL DASHON
    IF (Y0.LE.YMAX) GO TO 655
```

```
     STATEMENT


 654 CALL PLOT(XMAX,YMAX,2)
     CALL INTRST (0.,YMAX,XMAX,YMAX,0.,0.,X,Y,X0,Y0)
 655 CALL PLOT(X0,Y0,2)
     CALL DASHOF
     RETURN
 653 IF (Y.GT.YMAX.AND.YP.GT.YMAX) GO TO 656
     GO TO 657
 656 CALL DASHON
     CALL INTRST (0.,YMAX,XMAX,YMAX,0.,0.,X,Y,X0,Y0)
     IF (X0.LE.XMAX) GO TO 659
 658 CALL INTRST (XMAX,0.,XMAX,YMAX,0.,0.,X,Y,X0,Y0)
 660 CALL PLOT(X0,Y0,2)
     CALL DASHOF
 657 RETURN
 659 IF(IAXIS.EQ.1) CALL PLOT(XMAX,YMAX,2)
     GO TO 660
 806 CONTINUE
     GOTO 809
 807 PRINT 808
 808 FORMAT(1H1,' END OF FILE HAS BEEN READ WHERE DATA CARD SHOULD BE')
809    END
```

```
      STATEMENT


      SUBROUTINE PDSN(PDT,PFA,NPULS,KASE,SDB)
      EXTERNAL PD
      DIMENSION DB0(5),SLOPE(5),PDFAC(5)
      COMMON /PDS/ PFLAST,NLAST,KSLAST
      DATA PFLAST/0./
      DATA NLAST/0/
      DATA KSLAST/-1/
      DATA DB0/12.5,14.,14.,13.2,13.2/
      DATA SLOPE/6.,7.,8.,7.,8./
      DATA PDFAC/4.8,20.,20.,13.,13./
      DATA DBMIN/-30./
      DATA DBMAX/50./
      DATA DB1/0./
      DATA DB2/6./
      DATA PDLAST/0./
      IF (PDT.NE.PDLAST) GO TO 20
    1 IF (PFA.NE.PFLAST) GO TO 20
    2 IF (NPULS.NE.NLAST) GO TO 20
    3 IF (KASE.EQ.KSLAST) RETURN
   20 PDLAST=PDT
      PFLAST=PFA
      NLAST=NPULS
      KSLAST=KASE
      K=KASE + 1
      PULS=NPULS
      DB1=DB0(K)-SLOPE(K)*ALOG10(PULS)+(PDT-.5)*PDFAC(K)+(PFA-8.)*.4-1.
      DB2=DB1+2.
      CALL INVERS(DBMIN,DBMAX,DB1,DB2,4,15,NSI,SDB,PD1,PDT,PD)
      END
```

```
STATEMENT


FUNCTION PD(SNDB)
COMMON/PDS/FA,N,KASE
NP=N
FAN=FA
KAS=KASE
CALL MARSAR (SNDB,NP,FAN,KAS,PD1)
PD=PD1
END
```

```
    STATEMENT


    SUBRØUTINE INVERS(XMIN,XMAX,XLØ,XHI,NSIG,LIM,NØI,X,F1,FT,F)
    TEST = 10.**(-NSIG)
    FD = FT
    IF (FT .EQ. 0.) FD = 1.
    NØI = 1
    DELTA = XHI - XLØ
    X1=XLØ
    X2=XHI
    F1=F(X1)
    F2=F(X2)
    SLØPE=(F2-F1)/(X2-X1)
    IF (SLØPE.NE.0.) GØ TØ 21
 10 FMAX=F(XMAX)
    FMIN=F(XMIN)
    SLØPE = (FMAX-FMIN)/(XMAX-XMIN)
 21 IF ((F2-FT)*SLØPE.GE.0.) GØ TØ 23
 22 X1=X2
    F1=F2
    X2=X2+DELTA
    IF (X2 .GT. XMAX) Y2=XMAX
    F2=F(X2)
    GØ TØ 21
 23 IF ((FT-F1)*SLØPE.GE.0.) GØ TØ 25
 24 X2=X1
    X1=X1-DELTA
    IF (X1 .LT. XMIN) X1=XMIN
    F2=F1
    F1=F(X1)
    GØ TØ 23
 25 XA=X1
    XB=X2
    FB=F2
    IF (ABS (F2-FT).GE.ABS (F1-FT)) GØ TØ 6
  7 F22=F2
    F2=F1
    F1=F22
    X22=X2
    X2=X1
    X1=X22
    GØ TØ 6
  1 F1=F(X)
    X1=X
    TEST1 = ABS((F1-FT)/FD)
    IF (TEST1.GT.TEST) GØ TØ 6
  2 RETURN
  6 IF(NØI.LT.LIM) GØ TØ 13
 12 PRINT 40
    PRINT 41, LIM
    PRINT 42, XMIN, XMAX, XLØ, XHI, NSIG, LIM, NØI, X, F1, FT
    RETURN
 13 IF (F1.NE.F2) GØ TØ 16
```

```
   STATEMENT

15 IF (F1.NE.FB) GO TO 18
17 XB=X1
   X2=XB
   GO TO 19
18 XA=X1
   X2=XA
19 X=(XA+XB)*.5
   GO TO 1
16 X=(X1-X2)*(FT-F2)/(F1-F2) + X2
   IF (X .LT. XA) X=XA
   IF (X .GT. XB) X=XB
   NOI = NOI + 1
   F2=F1
   X2=X1
   GO TO 1
40 FORMAT (//'   MESSAGE FROM SUBROUTINE INVERS -- '//)
41 FORMAT ('   FUNCTION INVERSION NOT ACCOMPLISHED WITHIN SPECIFIED '
  *  ,I3, ' ITERATIONS.'/)
42 FORMAT (/'   INVERS PARAMETERS WERE ',4(E10.3,2X),3(I3,2X),E10.3,
  *  2(2X,E10.3) //)
   END
```

```
    STATEMENT


    SUBROUTINE MAPSWR (SNDB,N,FA,KASE,PN)
    DOUBLE PRECISION ENPR,YHPR,GAMPR,PYB,H,Y0,E0,Y1,E1,STEP,YB
    DOUBLE PRECISION DGAM, DEVAL, SUMLOG, SUML, FAN, EN
    SNR = 10.**(SNDB*.1)
    MODE=1
    IF (MODE) 800, 800, 900
900 FAN=DLOG10(DLOG(0.5D0)/DLOG(1.-(10.D0)**(-FA)))
    GO TO 905
800 FAN = FA
905 IF(N) 99,99,2
  2 IF(FA)99,99,3
  3 IF(KASE) 99,4,4
  4 IF(KASE-4) 5,5,99
  5 ENPR = 0.
  6 ENPR = FAN
    EN = N
    YBPR = 0.
    IF (NPREV .EQ. N .AND. FAPREV .EQ. FA) GO TO 777
    IF(N-12) 7,7,8
  7 YBPR=EN*(1.+2.2*ENPR/EN**((2.D0/3.D0)+0.015*ENPR))
    GO TO 11
  8 YBPR = EN*(1.+1.3*ENPR/EN**(.5+.011*ENPR))
 11 ENPR = 10.**ENPR
    GAMPR = DGAM(YBPR,N-1)
    PYB = .5**(1./ENPR)
    SUML = SUMLOG(N-1)
    IF(GAMPR-PYB) 10,12,12
 10 H = .01
    GO TO 14
 12 H = -.01
 14 Y0 = YBPR
    E0 = DEVAL(Y0,N-1,SUML)
 16 Y1 = Y0+H
    E1 = DEVAL(Y1,N-1,SUML)
    STEP = GAMPR + H*(E0+E1)/2.
    IF ((DSIGN(1.D0,STEP-PYB)-DSIGN(1.D0,H)).EQ.0.) GO TO 20
 18 Y0 = Y1
    E0 = E1
    GAMPR = STEP
    GO TO 16
 20 IF(H) 22,24,24
 22 YB = Y1 - H*(PYB-STEP)/(GAMPR-STEP)
    GO TO 30
 24 YB = Y0 + H*(PYB-GAMPR)/(STEP-GAMPR)
 30 BIAS = YB
777 YB = BIAS
    NPREV = N
    FAPREV = FA
    X = SNR
    K = KASE+1
    GO TO (100,200,300,400,500), K
    STATEMENT


100 SUM = 0.
    P = EN*X
    IF(YB-P-EN) 150.102,102
102 KS = -(EN+1.)/2. + DSQRT(((EN-1.)/2.)**2+P*YB)
    KS =  MAX0 (KS,0)
    GS = 1.-GAM(YB,KS+N-1,TN)
    TS = EVAL(P,KS)*GS
    G = GS
    K = KS
    TERM = TS
    TL = TN
110 TEMP = SUM+TERM
    IF(SUM-TEMP) 112,116,116
112 SUM = TEMP
    IF(K) 115.116,114
114 TERM = TERM*FLØAT (K)*(G-TL)/(P*G)
    G = G-TL
    K = K-1
    TL = TL*FLØAT (K+N)/YB
    GØ TØ 110
116 TL = TN*YB/FLØAT (KS+N)
    K = KS+1
    G = GS+TL
    TERM = TS*P*G/(GS*FLØAT (K))
120 TEMP = SUM+TERM
    IF(SUM-TEMP) 122,190,190
122 SUM = TEMP
    TL = TL*YB/FLØAT (K+N)
    K = K+1
    TERM = TERM*P*(G+TL)/(G*FLØAT (K))
    G = G+TL
    GØ TØ 120
150 KS = -1. - EN/2. + DSQRT(EN**2/4.+P*YB)
    KS = MAX0 (KS,0)
    GS = GAM(YB,KS+N-1,TN)
    IF(GS) 174,174,155
155 TS = EVAL(P,KS)*GS
    G = GS
    TERM = TS
    K = KS
    TL = TN
160 TEMP = SUM+TERM
    IF(SUM-TEMP) 162,166,166
162 SUM = TEMP
    IF(K) 165,166,164
164 TERM = TERM*FLØAT (K)*(G+TL)/(P*G)
    G = G+TL
    TL = TL*FLØAT (K+N-1)/YB
    K = K-1
    GØ TØ 160
166 TL = TN*YB/FLØAT (KS+N)
```

```
    STATEMENT


    K = KS+1
    G = GS-TL
    TERM = TS*P*G/(GS*FLOAT (K))
170 TEMP = SUM + TERM
    IF(SUM-TEMP) 172,174,174
172 SUM = TEMP
    TL = TL*YB/FLOAT (K+N)
    TERM = TERM*P*(G-TL)/(G*FLOAT (K+1))
    G = G-TL
    K = K+1
    GO TO 170
174 SUM = 1.-SUM
190 PN = SUM
    GO TO 90
200 IF(N-1) 210,210,220
210 PN = DEXP(-YB/(1.+X))
    GO TO 90
220 TEMP = 1. + 1./(EN*X)
    PN = 1. - GAM(YB,N-2,DUM) + DEXP((EN-1.)*ALOG(TEMP)-YB/(1.+EN*X))
   *                            *GAM(YB/TEMP,N-2,DUM)
    GO TO 90
300 IF(N-1) 310,310,320
310 PN = DEXP(-YB/(1.+X))
    GO TO 90
320 PN = 1. - GAM(YB/(1.+X),N-1,DUM)
    GO TO 90
400 IF(N-2) 410,420,430
410 PN = (1.+2.*X*YB/(X+2.)**2)*DEXP(-2.*YB/(2.+X))
    GO TO 90
420 PN = (1.+YB/(1.+X))*DEXP(-YB/(1.+X))
    GO TO 90
430 C = 2./(2.+EN*X)
    D = 1.-C
    IF(YB*D-EN) 440,450,450
440 SUM = 0.
    TERM = 1.
    J = N
442 TEMP = SUM+TERM
    IF(SUM-TEMP) 444,446,446
444 SUM = TEMP
    TERM = TERM*YB*D/FLOAT (J)
    J = J+1
    GO TO 442
446 PN = 1. - GAM(YB,N-2,DUM) + C*YB*EVAL(YB,N-2)
   *          + D*EVAL(YB,N-1)*(1.+C*YB-(EN-2.)*C/D)*SUM
    GO TO 90
450 PN = 1. - GAM(YB,N-3,DUM) + YB*EVAL(YB,N-3)*C/D
   *          + DEXP(-C*YB-(EN-2.)*ALOG(D))*(1.+C*YB-(EN-2.)*C/D)
   *           *GAM(YB*D,N-3,DUM)
    GO TO 90
500 SUM = 0.
```

```
      STATEMENT

      C = 2./(2.+X)
      D = 1.-C
      G = C/D
      P = C*YB
      KS = (3.*EN+(YB*D))/2.-DSQRT((EN-1.+(YB*D))**2/4.+(YB*D)*(EN+1.))
      KS =  MIN0 (KS,N)
      KS =  MAX0 (KS,0)
      K = KS
      J = N-KS
      FKS = KS
      K =  MIN0 (KS,N)
      IF(YB-EN*(1.+D)) 550,501,501
  501 GS = 1. - GAM(P,2*N-1-KS,TN)
      IF(GS) 526,526,502
  502 TS = DEXP(FKS*ALØG(C)+(EN-FKS)*ALØG(D)+SUMLØG(N)-SUMLØG(KS)
     *            -SUMLØG(J)+ALØG(GS))
      G = GS
      TERM = TS
      TL = TN
  510 TEMP = SUM+TERM
      IF(SUM-TEMP) 512,516,516
  512 SUM = TEMP
      IF(K) 516,516,514
  514 TL = TL*P/FLØAT (2*N-K)
      TERM = TERM*FLØAT (K)*(G+TL)/(G*FLØAT (N-K+1)*G)
      G = G+TL
      K = K-1
      GØ TØ 510
  516 IF(KS-N) 518,526,526
  518 TERM = TS*G*FLØAT (N-KS)*(GS-TN)/(FLØAT (KS+1)*GS)
      G = GS-TN
      TL = TN*FLØAT (2*N-1-KS)/P
      K = KS+1
  520 TEMP = SUM+TERM
      IF(SUM-TEMP) 522,526,526
  522 SUM = TEMP
      IF(K-N) 524,526,526
  524 TERM = TERM*G*FLØAT (N-K)*(G-TL)/(FLØAT (K+1)*G)
      G = G-TL
      TL = TL*FLØAT (2*N-1-K)/P
      K = K+1
      GØ TØ 520
  526 PN = SUM
      GØ TØ 90
  550 GS = GAM(P,2*N-1-KS,TN)
      IF(GS) 576,576,552
  552 TS = DEXP(FKS*ALØG(C)+(EN-FKS)*ALØG(D)+SUMLØG(N)-SUMLØG(KS)
     *           -SUMLØG(J)+ALØG(GS))
      G = GS
      TERM = TS
      TL = TN
```

```
STATEMENT


560 TEMP = SUM+TERM
    IF(SUM-TEMP) 562,566,566
562 SUM = TEMP
    IF(K) 566,566,564
564 TL = TL*P/FLØAT (2*N-K)
    TERM = TERM*FLØAT (K)*(G-TL)/(G*FLØAT (N-K+1)*G)
    G = G-TL
    K = K-1
    GØ TØ 560
566 IF(KS-N) 568,576,576
568 TERM = TS*Q*FLØAT (N-KS)*(GS+TN)/(FLØAT (KS+1)*GS)
    G = GS+TN
    TL = TN*FLØAT (2*N-1-KS)/P
    K = KS+1
570 TEMP = SUM+TERM
    IF(SUM-TEMP) 572,576,576
572 SUM = TEMP
    IF(K-N) 574,576,576
574 TERM = TERM*Q*FLØAT (N-K)*(G+TL)/(FLØAT (K+1)*G)
    G = G+TL
    TL = TL*FLØAT (2*N-1-K)/P
    K = K+1
    GØ TØ 570
576 PN = 1.-SUM
    GØ TØ 90
 90 IF(PN) 91,94,92
 91 PN = 0.
    GØ TØ 94
 92 IF(PN-1.) 94,94,93
 93 PN = 1.
 94 RETURN
 99 WRITE (61,9)  N,FA ,SNR,KASE
  9 FØRMAT (1H0 /50H UNREASØNABLE CALL SEGUENCE TØ MARCUM, ZERØ RESULT
   *                 7HS GIVEN //4H N = I8,5X,5HFA  = E16.8,5X,5HSNR =
   *                E16.8,5X,6HKASE = I8)
    PN = 0.
    BIAS = 0.
    RETURN
    END
```

```
     STATEMENT

      FUNCTIØN DGAM(B,N)
      DØUBLE PRECISIØN  SUM,  TERM,TEMP,FJ,DGAM, DEVAL, B, SUML,SUMLØG
      SUM = 0.
      K = B
      IF(K-N) 100,200,200
  100 J = N+1
      SUML = SUMLØG(J)
      TERM = DEVAL(B,J,SUML)
   10 TEMP = SUM+TERM
      IF(SUM-TEMP) 15,20,20
   15 SUM = TEMP
      J = J+1
      FJ = J
      TERM = TERM*B/FJ
      GØ TØ 10
   20 DGAM = SUM
      RETURN
  200 J = N
      SUML = SUMLØG(J)
      TERM = DEVAL(B,J,SUML)
   30 TEMP = SUM+TERM
      IF(SUM-TEMP) 35,40,40
   35 SUM = TEMP
      IF(J-1) 40,36,36
   36 FJ = J
      TERM = TERM*FJ/B
      J = J-1
      GØ TØ 30
   40 DGAM = 1.-SUM
      RETURN
      END
```

```
    STATEMENT

    FUNCTION DEVAL (Y,N,SUML)
    DOUBLE PRECISION  XPON,EN,DEVAL,  Y,SUML
    XPON = -Y
    IF(N) 20,20,10
 10 EN = N
    XPON = XPON+EN*DLOG(Y)-SUML
 20 DEVAL = DEXP(XPON)
    RETURN
    END
```

```
      STATEMENT


      FUNCTION GAM(B,N,TN)
      SUM = 0.
      K = B
      IF(K-N) 100,200,200
  100 J = N+1
      TERM = EVAL(B,J)
      TN = TERM*FLOAT (J)/B
   10 TEMP = SUM+TERM
      IF(SUM-TEMP) 15,20,20
   15 SUM = TEMP
      J = J+1
      FJ = J
      TERM = TERM*B/FJ
      GO TO 10
   20 GAM = SUM
      RETURN
  200 J = N
      TERM = EVAL(B,J)
      TN = TERM
   30 TEMP = SUM+TERM
      IF(SUM-TEMP) 35,40,40
   35 SUM = TEMP
      IF(J-1) 40,36,36
   36 FJ = J
      TERM = TERM*FJ/B
      J = J-1
      GO TO 30
   40 GAM = 1.-SUM
      RETURN
      END
```

```
    STATEMENT


    FUNCTIØN EVAL(Y,N)
    XPØN = -Y
    IF(N) 20,20,10
 10 EN = N
    XPØN = XPØN+EN*ALØG(Y)-SUMLØG(N)
 20 EVAL = EXP (XPØN)
    RETURN
    END
    STATEMENT


    FUNCTIØN SUMLØG(N)
    DØUBLE PRECISIØN A, B, SUMLØG
    DIMENSIØN A(1000)
    DATA DUMA / 0./
    DATA DUMB / 0./
    DATA NLAST/1/
    NMAX=1000
    IF(DUMA-DUMB) 20,10,20
 10 DUMA = 1.
    DUMB = 0.
    NLAST = 1
    A(1) = 0.
 20 NN = IABS (N)
    IF(NN-1) 30,30,40
 30 SUMLØG = 0.
    RETURN
 40 IF(NN-NLAST) 50,50,60
 50 SUMLØG = A(NN)
    RETURN
 60 K = NLAST+1
    IF(NN-NMAX) 70,70,80
 70 DØ 72 I=K,NN
 72 A(I) = A(I-1) + DLØG(DFLØAT(I))
    NLAST = NN
    GØ TØ 50
 80 IF(NLAST-NMAX) 82,90,90
 82 DØ 84 I=K,NMAX
 84 A(I) = A(I-1) + DLØG(DFLØAT(I))
    NLAST = NMAX
 90 B = A(NMAX)
    K = NMAX+1
    DØ 92 I=K,NN
 92 B = B + DLØG(DFLØAT(I))
    SUMLØG = B
    RETURN
    END
```

```
     STATEMENT

     SUBROUTINE RHACHT(XMAX,YMAX,RMAX,HMAX,ANTHGT)
     EXTERNAL F1
     DIMENSION SN(181), RNG1(181), TN1(181), JA(6)
     DIMENSION XZ(180),YZ(180)
     DIMENSION IRFAC (5), IHFAC (8)
     COMMON /A/ REF, GRAD, RAD, CONST, U(182), N
     COMMON /MTPE/ X22,Y22,X11,Y11,XAA,YAA,ERROR
     DIMENSION IANG(9),AANG(9)
     COMMON /B/ XX(181), YY(181), CT1(181), SN1(181), DEL
     DATA ERROR / .001/
     DATA REF/.000313/
     DATA GRAD/.00004384822/
     DATA IANG/1,26,31,51,71,121,151,172,181/
     DATA AANG/0.,.5,1.,3.,5.,10.,30.,60.,90./
     DATA IHFAC/50,100,500,1000,5000,10000,50000,100000/
     DATA IRFAC/5,10,50,100,500/
     DATA JA/31,51,71,121,151,172/
     CALL PENCHG(10)
     DEL=XMAX*.01
     IF (YMAX.LT.XMAX) DEL=YMAX*.01
     BA = 6076.1155*RMAX*YMAX/(HMAX*XMAX)
     BA2 = BA*BA
     PREC = .00001
     CONST = .3048/1852.0
     RAD=20898950.13+ANTHGT
     AB = 1.0 + REF
     AB2 = AB*AB
     CD = 2.0 * REF + REF*REF
     ELEV = -.02
     II=0
     DO 29 JI=1,6
     GO TO (490,491,492,493,494,495), JI
 490 ADEL = .02
     MM=26
     GO TO 51
 491 ADEL = .1
     MM=95
     GO TO 51
 492 ADEL=.5
     MM=20
     GO TO 51
 493 ADEL = 1.0
     MM=25
     GO TO 51
 494 ADEL=2.5
     MM=12
     GO TO 51
 495 ADEL=5.0
     MM=3
  51 DO 29 IK = 1,MM
     II=II+1
     STATEMENT


     ELEV=ELEV+ADEL
     RDN = ELEV/57.29577957
     SN(II) = SIN(RDN)
     S = SN(II)**2
     U(II) = AB2*S-CD
     IF (II.GE.181) GO TO 361
 360 TN = TAN (RDN)
     RDN1 = ATAN (BA*TN)
     TN1(II) = TAN (RDN1)
     CT1(II) = COS(RDN1)
     SN1(II) = SIN(RDN1)
     GO TO 29
 361 CT1(II) = 0.0
     SN1(II) = 1.0
  29 CONTINUE
     B=BA*XMAX
     DO 290 IX= 1,180
     BAT=SQRT(BA2+(TN1(IX))**2)
     XZ(IX)=XMAX*BA/BAT
 290 YZ(IX)=B*TN1(IX)/BAT
     H1 = 0.0
     IJM = 100
     JMAX = INT (ALOG10(HMAX) - .477122)
     HINT=10.**(JMAX-1)
     IJQ = (HMAX/(10.**(JMAX-1))+ .001)
     HMX=IJQ*10**(JMAX-1)
     DO 30 J = 1, JMAX
     IF (J .EQ. JMAX) IJM = IJQ
     IF (J.NE.1) GO TO 3
   2 IJ  = 10
     GO TO 4
   3 IJ = 20
   4 DO 30 I = IJ,IJM,10
     H2 = I * 10 ** (J - 1)
     IF(H2.EQ.HMX) CALL PENCHG(11)
     DO 304 K = 1, 181
     N=182-K
     IF (H2 .EQ. 10 .) RNG1(N) = 0.0
     IF (H2.EQ.10..AND.N.EQ.1) GO TO 6
     GO TO 7
   6 GAM = REF*GRAD/AB
     RI = 1.0/RAD
     GG = 2;0*(RI - GAM)
     RNG2 = (CONST*AB/GG)*2.0*SQRT(GG*H2)
     GO TO 8
   7 CALL SIMCON (H1,H2,PREC  ,15,RINC,NO1,R,F1)
     RNG2 = RNG1(N) + RINC
   8 RNG1(N) = RNG2
     IF (H2 .LT. HINT) GO TO 304
     A = RNG2*XMAX/RMAX
     B = BA*A
```

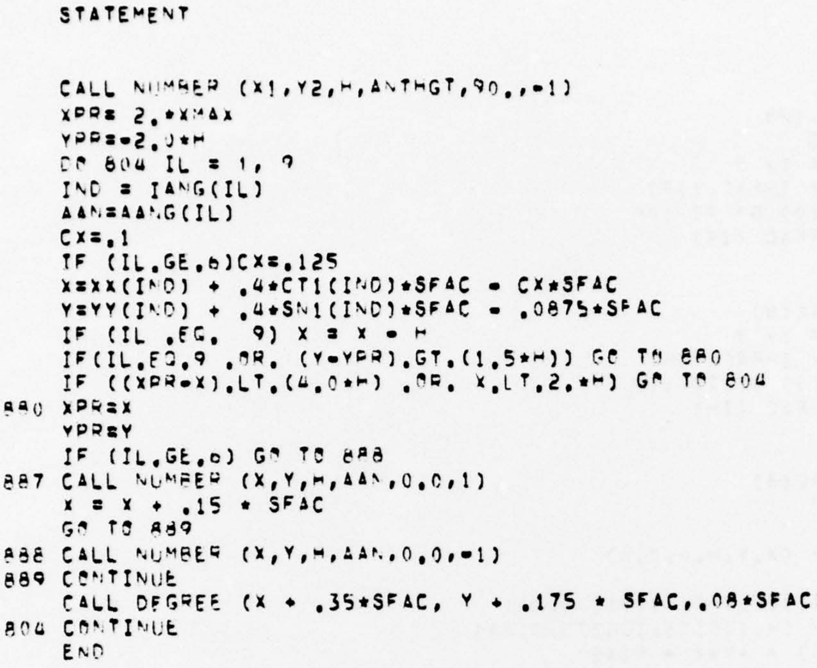
STATEMENT

```
    IF (N.NE.181) GO TO 47
 46 CALL PLOT (0.,8,3)
    XX(181) = 0.
    YY(181) = 8
    X=0.
    Y=8
    X22=0.
    Y22=8
    GO TO 304
 47 XLAST = X
    YLAST = Y
    X=A*XZ(N)/XMAX
    Y=A*YZ(N)/XMAX
 48 IF (H2.NE.HMAX) GO TO 68
 87 XX(N)=X
    YY(N)=Y
 68 IF(K.NE.2) GO TO 781
780 X11=X
    Y11=Y
    XAA=X
    YAA = Y
    GO TO 304
781 IF (X.LE.(XZ(N)+.0001)) GO TO 783
782 CALL INTRST (XLAST,YLAST,X,Y,XZ(N),YZ(N),XZ(N+1),YZ(N+1),X0,Y0)
    CALL MINTAP (X0,Y0)
    CALL PLOT (X0,Y0,2)
    IF (H2.EQ.HMAX) GO TO 785
    GO TO 305
783 IF (K.NE.181) GO TO 788
787 CALL MINTAP (X,Y)
    CALL PLOT(X,Y,2)
    XCOR = X
    YCOR = Y
    GO TO 304
788 CALL MINTAP (X,Y)
304 CONTINUE
305 H1 = H2
 30 CONTINUE
    GO TO 789
785 CALL PLOT(XZ(1),0.,3)
    DO 786 NK=1,N
    XX(NK)=XZ(NK)
    YY(NK)=YZ(NK)
786 CALL PLOT(XX(NK),YY(NK),2)
    CALL PLOT(X0,Y0,2)
    XCOR = X0
    YCOR = Y0
789 X=0.
    Y = YMAX
    CALL PLOT(X,Y,3)
    Y = 0.0
```

```
      STATEMENT


      CALL PLOT(X,Y,2)
      X=XMAX
      CALL PLOT(X,Y,2)
      CALL PENCHG(10)
      KAM=RMAX-1.
      INTR=10
      IF(RMAX.LT. 100.) INTR=5
      IF(RMAX.GT. 300.) INTR=25
      DO 31 KA=INTR,KAM,INTR
      RG = KA
      A = RG*XMAX/RMAX
      B = BA+A
      DO 38 KC = 1, 181
      KR=KC
      IF (KC.NE.181) GO TO 92
   91 CALL MINTAP (0.,B)
      CALL PLOT(0.,B,2)
      GO TO 31
   92 X=A*XZ(KC)/XMAX
      IF (X.GT.XX(1)) GO TO 31
      Y=A*YZ(KC)/XMAX
      IF (Y.LE.(YY(KC)+.0001)) GO TO 72
   73 IF (KR.NE.N+1) GO TO 731
  730 X1=XCOR
      Y1=YCOR
      GO TO 732
  731 X1=XX(KR-1)
      Y1 = YY(KR-1)
  732 X2=XX(KR)
      Y2 = YY(KR)
      XB=X
      YB = Y
      CALL INTRST (X1,Y1,X2,Y2,XA,YA,XB,YB,X0,Y0)
      CALL PLOT(X0,Y0,2)
      GO TO 31
   72 XA = X
      YA = Y
      IF (KC.NE.1) GO TO 86
   85 CALL PLOT(X,Y,3)
      X22=X
      Y22=Y
      GO TO 38
   86 IF (KC.NE.2) GO TO 861
  860 X11=X
      Y11=Y
      XAA=X
      YAA=Y
      GO TO 38
  861 CALL MINTAP (X,Y)
   38 CONTINUE
   31 CONTINUE
```

```
    STATEMENT

700 CALL ATICK(1,26,5,4,5)
    CALL ATICK(27,71,1,5,10)
    CALL ATICK(73,141,2,0,5)
    CALL ATICK(142.151,1,0,5)
    CALL ATICK(156,166,5,0,2)
    CALL ATICK(168,178,2,1,2)
    CALL ATICK(179,181,1,1,2)
    DO 34 KF = 1, 6
    NF = JA(KF)
    X = 0.0
    Y = 0.0
    CALL PLOT(X,Y,3)
    X = XX(NF)
    Y = YY(NF)
    CALL PLOT(X,Y,2)
 34 CONTINUE
    KT = 9
    DO 364 KY = 1, 800
    KT = KT + 1
    IF (KT.NE.10) GO TO 369
368 FAC = 2.0
    KT = 0
    GO TO 370
369 FAC = 1.0
370 R2 = KY - 1
    X = R2*XMAX/RMAX
    IF (X .GT. (XMAX + .0001)) GO TO 365
    Y = 0.0
    CALL PLOT(X,Y,3)
    Y = - FAC * DEL
    CALL PLOT(X,Y,2)
364 CONTINUE
365 KS = 9
    KJM = HMAX / HINT + 1.001
    DO 37 KJ = 1, KJM
    KS = KS + 1
    IF (KS.NE.10) GO TO 376
375 FAC = 2.0
    KS = 0
    GO TO 377
376 FAC = 1.0
377 H = HINT * (KJ -1)
    Y = H*YMAX/HMAX
    X = 0.0
    CALL PLOT(X,Y,3)
    X = - FAC * DEL
    CALL PLOT(X,Y,2)
 37 CONTINUE
    CALL PENCHG(11)
    IF (XMAX-YMAX) 460,460,461
460 SFAC=XMAX +.125
```

```
     STATEMENT


     GO TO 402
 401 SFAC=YMAX*.125
 402 H=.175*SFAC
     DO 100 IR = 1, 5
     NR = RMAX / IRFAC (IR)
     IF (NR.GT.10) GO TO 100
 101 IRUNIT = IRFAC (IR)
     GO TO 102
 100 CONTINUE
     IRUNIT=IRFAC(5)
 102 DO 110 IH = 1, 8
     NH = HMAX / IHFAC (IH)
     IF (NH.GT.10) GO TO 110
 103 IHUNIT = IHFAC (IH)
     GO TO 128
 110 CONTINUE
     IHUNIT=IHFAC(8)
 128 X=-.05*SFAC
     Y=-.5*SFAC
     CALL INUMBR (X,Y,H,0,0.0)
     N = IRUNIT
 120 IDGITS=ALOG10(FLOAT(N))+1.000001
     CALL SENTER (H,IDGITS,IDGITS,BIAS)
     X = (N/RMAX) * XMAX - BIAS
     IF (X + BIAS .GT. XMAX) GO TO 801
     CALL INUMBR (X,Y,H,N,0.0)
     N = N + IRUNIT
     GO TO 120
 801 Y=-1.0*SFAC
     CALL SENTER (H, 21,21,BIAS)
     X = 0.5 * XMAX - BIAS
     CALL LETTER (X,Y,H,21HRANGE, NAUTICAL MILES,0.0,21)
     X=-.5*SFAC
     Y=-.0875*SFAC
     CALL INUMBR (X,Y,H,0,0.0)
     N = IHUNIT
 121 IHGITS=ALOG10(FLOAT(N))+3.000001
     X=-0.15*IHGITS*SFAC
     Y=(N/HMAX)*YMAX-.0875*SFAC
     IF (Y .GT. YMAX) GO TO 803
     CALL INUMBR (X,Y,H,N,0.0)
     N = N + IHUNIT
     GO TO 121
 803 X = -1.40*SFAC
     CALL SENTER(H,33,33,BIAS)
     Y = 0.5 * YMAX - BIAS
     CALL LETTER (X,Y,H,33HTARGET HEIGHT ABOVE ANTENNA, FEET,90.,33)
     X1=X-3.*H
     CALL SENTER(H,26,26,BIAS)
     CALL LETTER (X1,Y,H,26H(ANTENNA HEIGHT=      FEET),90.,26)
     Y2=Y+10.5714*H
```



```
     STATEMENT

     CALL NUMBER (X1,Y2,H,ANTHGT,90.,-1)
     XPR= 2.*XMAX
     YPR=-2.0*H
     DO 804 IL = 1, 9
     IND = IANG(IL)
     AAN=AANG(IL)
     CX=.1
     IF (IL.GE.6)CX=.125
     X=XX(IND) + .4*CT1(IND)*SFAC - CX*SFAC
     Y=YY(IND) + .4*SN1(IND)*SFAC - .0875*SFAC
     IF (IL .EQ.  9) X = X - H
     IF(IL.EQ.9 .OR. (Y-YPR).GT.(1.5*H)) GO TO 880
     IF ((XPR-X).LT.(4.0*H) .OR. X.LT.2.*H) GO TO 804
 880 XPR=X
     YPR=Y
     IF (IL.GE.6) GO TO 888
 887 CALL NUMBER (X,Y,H,AAN,0.0,1)
     X = X + .15 * SFAC
     GO TO 889
 888 CALL NUMBER (X,Y,H,AAN,0.0,-1)
 889 CONTINUE
     CALL DEGREE (X + .35*SFAC, Y + .175 * SFAC,.08*SFAC)
 804 CONTINUE
     END
```

```
STATEMENT


FUNCTIØN F1(X)
CØMMØN /A/ REF, GRAD, RAD, CØNST, U(162), N
BB=REF*EXP(-GRAD*X)
CC=X/RAD
V=2.0*BB+BB*BB
W=2.0*CC+CC*CC
FX = SQRT (U(N) + V + W + V*W)
F1 = CØNST * (1.0 +V)*(1.0 + CC)/FX
END
```

```
      STATEMENT


      SUBRØUTINE SIMCØN(X1,XEND,TEST,LIM,AREA,NØI,R,F)
      NØI=0
      ØDD=0.0
      INT=1
      V=1.0
      EVEN=0.0
      AREA1=0.0
19    ENDS=F(X1)+F(XEND)
2     H=(XEND-X1)/V
      ØDD=EVEN+ØDD
      X=X1+H/2.
      EVEN=0.0
      DØ 3 I=1,INT
21       EVEN=EVEN+F(X)
         X=X+H
3     CØNTINUE
31    AREA=(ENDS+4.0*EVEN+2.0*ØDD)*H/6.0
      NØI=NØI+1
34    R=ABS ((AREA1-AREA)/AREA)
      IF(NØI-LIM)341,35,35
341   IF(R-TEST)35,35,4
35    RETURN
4     AREA1=AREA
46    INT=2*INT
      V=2.0*V
      GØ TØ 2
      END
```

```
  STATEMENT


  SUBROUTINE ATICK (IA,JA,KA,MB,MC)
  COMMON /B/ XX(181), YY(181), CT1(181), SN1(181), DEL
  MA = MB
  DO 1 K = IA,JA,KA
  MA = MA + 1
  IF (MA.NE.MC) GO TO 3
2 FAC = 2.0
  MA = 0
  GO TO 4
3 FAC = 1.0
4 X = XX(K)
  Y = YY(K)
  CALL PLOT(X,Y,3)
  X = X + DEL*FAC*CT1(K)
  Y = Y + DEL*FAC*SN1(K)
  CALL PLOT(X,Y,2)
1 CONTINUE
  END
```

```
STATEMENT

SUBROUTINE SENTER (W,N,I,BIAS)
CONST = .2857143 * W
WIDTHF = (3*I-1) * CONST
WIDTHN = (3*N-1) * CONST
BIAS = WIDTHF - 0.5 * WIDTHN
RETURN
END
```

STATEMENT

```
      SUBROUTINE MINTAP    (X,Y)
      COMMON /MTPE/ X2, Y2, X1, Y1, XA, YA, ERROR
      COMMON /MN/ M,N
      DATA M/0/
      DATA N/0/
      DX(U1,V1,U2,V2) = SQRT((U2-U1)**2 + (V2-V1)**2)
      D1 = DX(X1,Y1,X2,Y2)
      IF (D1 .EQ. 0.) GO TO 11
      D2 = DX(X,Y,X1,Y1)
      IF (D2 .EQ. 0.) GO TO 2
      D3 = DX(X,Y,X2,Y2)
      D11 = DX(XA,YA,X2,Y2)
      IF (D3 .LT. D11) GO TO 1
      COSN = (D3*D3-D2*D2-D1*D1)/(2.*D1*D2)
      IF (COSN .GT. 1. .OR. COSN .LT. -1.) COSN = 1.
      SINE = SQRT(1. - COSN*COSN)
      DEVN = D2*SINE
      IF (DEVN .LE. ERROR) GO TO 2
    1 CALL PLOT (XA,YA,2)
      M = M+1
      X2 = XA
      Y2 = YA
   11 X1 = X
      Y1 = Y
    2 XA = X
      YA = Y
      N = N+1
      END
    STATEMENT


    SUBROUTINE SEAREF (FMHZ, PSI, IPOL, RHO, PHI)
          COMPLEX EPSC, GAM, SQTERM, TERM
    DATA FLAST / 0./
    SINPSI = SIN (PSI)
    CSPSI = COS(PSI)**2
    IF (FMHZ .EQ. FLAST) GO TO 200
    FLAST = FMHZ
    W = 299.793 / FMHZ
    IF (FMHZ .GT. 1500.) GO TO 151
150 SIG = 4.3
    EPS1 = 80.
    GO TO 155
151 SIG = 4.3 + (FMHZ - 1500.) * .00148
    IF (FMHZ .GT. 3000.) GO TO 154
153 EPS1 = 80. - (FMHZ - 1500.) * .00733
    GO TO 155
154 EPS1 = 69. -(FMHZ - 3000.) * .002429
    SIG = 6.52 + (FMHZ - 3000.) * .001314
155 EPSC = CMPLX (EPS1,-60.*W*SIG)
200 SQTERM= CSQRT(EPSC-CSPSI)
    IF (IPOL .NE. 1 ) GO TO 161
160 TERM =     EPSC * SINPSI
    GAM = (TERM-SQTERM)/(TERM+SQTERM)
    GO TO 180
161 GAM = (SINPSI - SQTERM) / (SINPSI + SQTERM)
180 RHO = CABS (GAM)
    PHI =-ATAN2 (AIMAG(GAM), REAL (GAM))
    RETURN
    END
```

STATEMENT

```
  SUBROUTINE DEGREE(X,Y,H)
  DIMENSION DELX(8), DELY(8)
  DATA DELX/-0.7071,-1.0,-0.7071,0.0,0.7071,1.0,0.7071,0.0/
  DATA DELY/0.7071,0.0,-0.7071,-1.0,-0.7071,0.0,0.7071,1.0/
  D = H*.414214
  X = X + .5*H
  Y = Y + .5*D
  CALL PLOT (X,Y,3)
  DO 1 II = 1, 8
  X = X + DELX(II)*D
  Y = Y + DELY(II)*D
1 CALL PLOT(X,Y,2)
  END
```

```
      STATEMENT


      SUBROUTINE INTRST  (X1,Y1,X2,Y2,XA,YA,XB,YB,X0,Y0)
      IF (ABS(X2-X1).GE.10.E-76) GO TO 2
    1 IF (ABS(XB-XA).LT.10.E-76) GO TO 99
      X0 = X1
      S = (YB-YA)/(XB-XA)
      Y0 = S*(X0-XA) + YA
      RETURN
    2 IF (ABS(XB-XA).GE.10.E-76) GO TO 4
    3 IF (ABS(X2-X1).LT.10.E-76) GO TO 99
      X0 = XA
      S = (Y2-Y1)/(X2-X1)
      Y0 = S*(X0-X1) + Y1
      RETURN
    4 S1 = (YB-YA)/(XB-XA)
      S2 = (Y2-Y1)/(X2-X1)
      IF (S1 .EQ. 0.) GO TO 5
    . IF (ABS(S2-S1).LT.10.E-76) GO TO 99
      RTO = S2/S1
      Y0 = (Y2+S2*(XA-X2)-YA*RTO)/(1.-RTO)
      X0 = (Y0-YA)/S1 + XA
      RETURN
    5 IF (S2 .EQ. 0.) GO TO 99
      X0= (YA-Y2)/S2 + X2
      Y0=YA
      RETURN
   99 PRINT 100, X1, Y1
      PRINT 101,     X2,Y2,XA,YA,XB,YB
      X0 = 0.
      Y0 = 0.
  100 FORMAT (3X,64HCALL TO INTRSECT ABORTED. LINES PARALLEL, NO INTERSE
     *CTION. X1 = ,E10.2,6H Y1 = ,E10.2)
  101 FORMAT(3X,                            6H X2 = ,E10.2,6H Y2 = ,E10.2,
     * 6H XA = ,E10.2,6H YA = ,E10.2,6H XB = ,E10.2,6H YB = ,E10.2 //)
      END
```

STATEMENT

```
SUBROUTINE ARROW (X1,Y1,X2,Y2,TIPL)
COMMON/AG/ANG
DATA ANG / .35/
YA = Y2 - Y1
XA = X2 - X1
A = ATAN2(YA,XA)
A3 = A + ANG
A4 = A - ANG
X3 = X2 - TIPL * COS(A3)
Y3 = Y2 - TIPL * SIN(A3)
X4 = X2 - TIPL * COS(A4)
Y4 = Y2 - TIPL * SIN(A4)
CALL PLOT (X1, Y1, 3)
CALL PLOT (X2, Y2, 2)
CALL PLOT (X3. Y3, 2)
CALL PLOT (X4, Y4, 3)
CALL PLOT (X2, Y2, 2)
END
```

STATEMENT

```
SUBROUTINE REZERO
ENTRY   PENCHG
ENTRY   DASHON
ENTRY   DASHOF
END
```

APPENDIX B: Input Data Card Formats for LOBMUF (For ASC - Jan 1977)

Program LOBMUF will plot the envelope of N radars using the pattern calculations contained in the original program described in NRL Report 7098 by L. Blake. Incoherent integration is assumed. An antenna height must be specified in subroutine RHACHT. The parameter name for this antenna height is ANTHITE and is used to draw the range-height grid. The actual antenna heights for the individual radars are input separately and used for the multipath calculations. ANTHITE is not very important except for the labeling of the height axis of the grid.

Input Card Sequence

		Col.
1.	ANTHITE in F10.0 field	
2.	Label Card	
3.	Grid Parameters in F10.0 fields:	
a.	XMAX	1-10
b.	YMAX	11-20
c.	RMAX	21-30
d.	HMAX	31-40
e.	THMIN	41-50
f.	THMAX	51-60
g.	WHFT	61-70
h.	RDR	71-80

XMAX = maximum X-dimension of chart in inches
YMAX = maximum Y-dimension of chart in inches
RMAX = maximum range of chart in nautical miles
HMAX = maximum height on chart in feet
THMIN = minimum elevation angle for the plot
THMAX = maximum elevation angle for the plot
WHFT = sea wave height in feet
RDR = number of radars for which an envelope is to be plotted.

4.	Common Parameters in F10.0 fields:	
a.	PDT	1-10
b.	PFA	11-20
c.	PULS (total number of pulses)	21-30
d.	CASE	31-40
e.	AHFT	41-50
f.	TILT	51-60
g.	POL	61-70
h.	CSC	71-80

PDT = probability of detection
PFA = false alarm exponent, i.e., the positive value of the exponent (power of ten); for 10^{-6}, enter the number 6.0, etc.
PULS = total number of pulses

CASE = Swerling case number
AHFT = antenna height in feet
TILT = tilt angle of the antenna beam maximum with respect to the horizon in degrees.

For reference, POL is polarization as follows:

POL = 1.	vertical
POL = 2.	horizontal

CSC is for pencil or cosecant squared antenna pattern:

CSC = 0.	pencil beam
CSC = 1.	cosecant squared beam

5. Radar Parameters in F10.0 fields:

a.	RFS	1-10
b.	FMHZ	11-20
c.	BWD	21-30
d.	SLDB	31-40
e.	PULNUM (number pulses at FMHZ)	41-50
f.	FREF	51-60

Repeat card 5 for N repetitions.

RFS = calculated or assumed free-space range of the radar on the specified target
FMHZ = radar frequency in megahertz
BWD = antenna half-power beamwidth in degrees
SLDB = first elevation sidelobe level relative to the main lobe
PULNUM = number of pulses at each specified frequency
FREF = frequency used to calculate RFS if RFS was calculated for the total number of pulses (PULS)
FREF = 0, if RFS was calculated at each frequency using PULNUM.

After cards 1-5, the sequence can be repeated as many times as desired, starting with card 2.

A note of caution is that the elevation plot angle increment is determined by

$$\frac{\text{THMAX} - \text{THMIN}}{2000}$$

The value has proven adequate for S-band radars where 10° and 0° were the values for THMAX and THMIN. It is recommended that an envelope for a single radar (RDR = 1) be run when there is doubt that this angle increment is small enough.